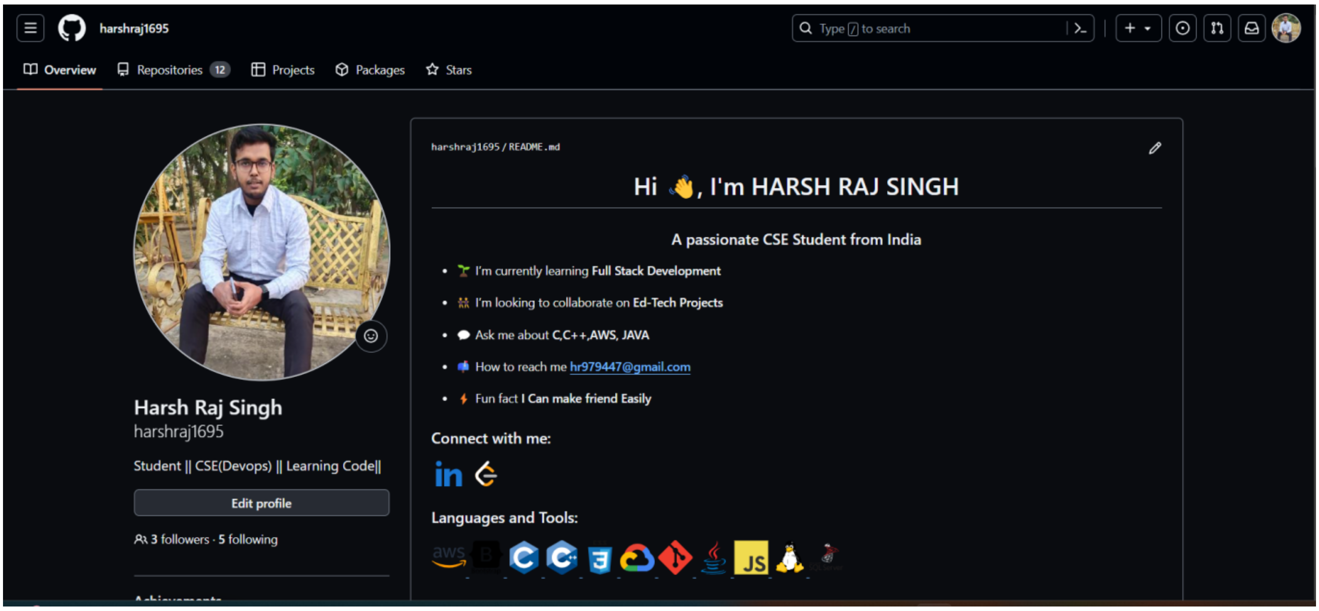
Assignment - 1

Student Name: HARSH RAJ SINGH **UID:** 22BDO10006
Branch: CSE (DevOps) **Section/Group-** 22BCD-1/A
Semester: 4th **Date of Submission:** 26/02/2024
Subject Name: Git and hub **Subject Code:** 22CSH-293

Task 1: Create Repository for agile, git and DAA practile files ad add a readme file to describe about each project.

Steps are following:

1. Go to your github account Create repository over GitHub.



harshraj1695 / README.md

Hi 👋, I'm HARSH RAJ SINGH

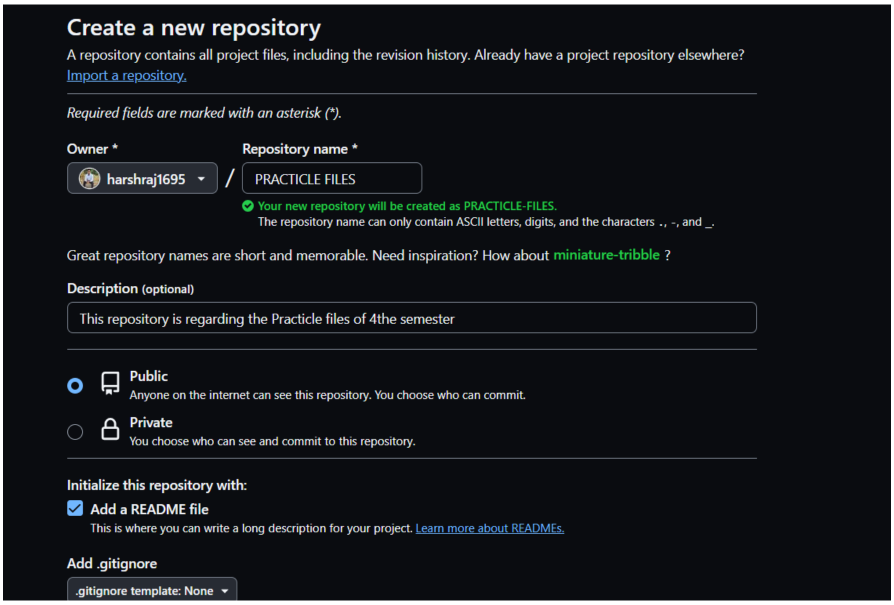
A passionate CSE Student from India

- 🌱 I'm currently learning **Full Stack Development**
- 👯 I'm looking to collaborate on **Ed-Tech Projects**
- 💬 Ask me about **C,C++,AWS, JAVA**
- 📫 How to reach me hr979447@gmail.com
- ⚡ Fun fact **I Can make friend Easily**

Connect with me:

Languages and Tools:

Harsh Raj Singh
harshraj1695
Student || CSE(Devops) || Learning Code||



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * harshraj1695 / Repository name * PRACTICLE FILES

✓ Your new repository will be created as PRACTICLE-FILES.
The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

Great repository names are short and memorable. Need inspiration? How about miniature-tribble ?

Description (optional)
This repository is regarding the Practicle files of 4the semester

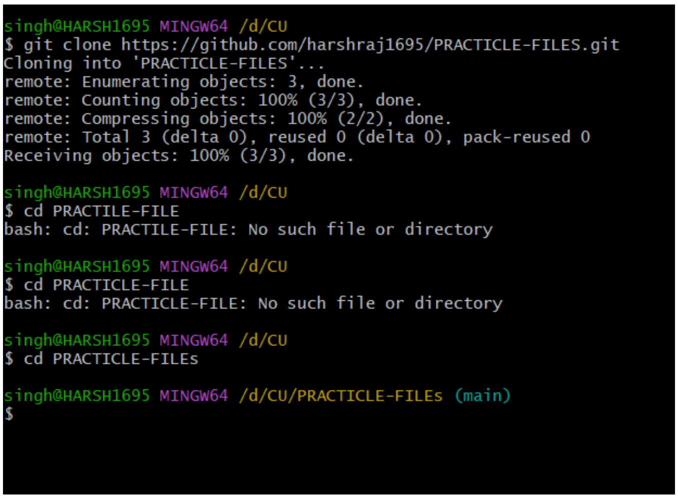
● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:
☑ Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore
.gitignore template: None

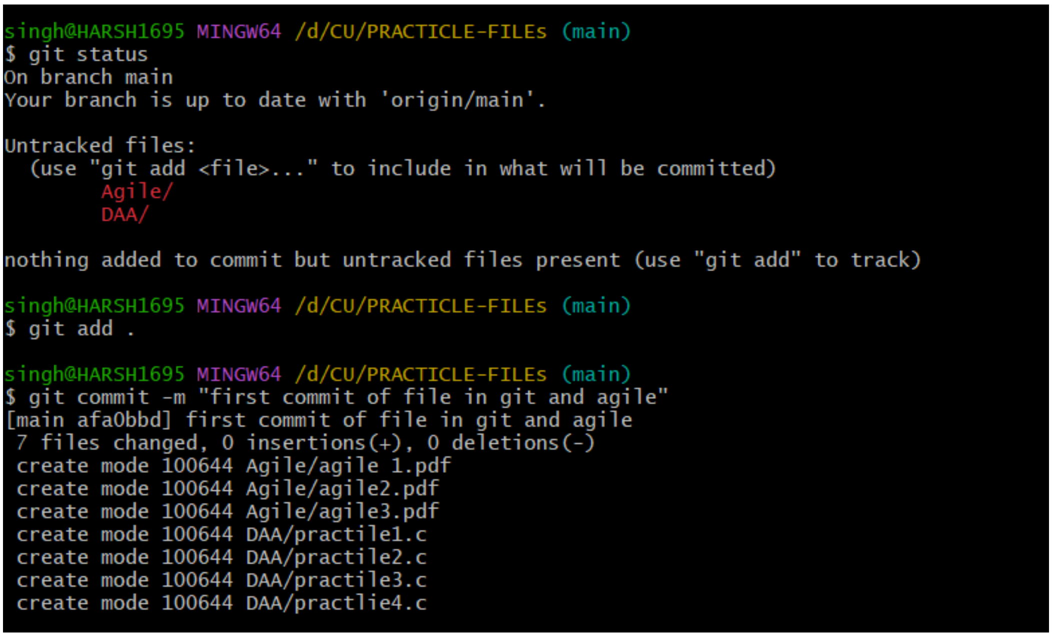
2. Now Clone this repository into the local machine and move to the repository using **“”cd ”** command.



```
singh@HARSH1695 MINGW64 /d/CU
$ git clone https://github.com/harshraj1695/PRACTICLE-FILES.git
Cloning into 'PRACTICLE-FILES'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

singh@HARSH1695 MINGW64 /d/CU
$ cd PRACTILE-FILE
bash: cd: PRACTILE-FILE: No such file or directory

singh@HARSH1695 MINGW64 /d/CU
$ cd PRACTICLE-FILE
bash: cd: PRACTICLE-FILE: No such file or directory

singh@HARSH1695 MINGW64 /d/CU
$ cd PRACTICLE-FILEs

singh@HARSH1695 MINGW64 /d/CU/PRACTICLE-FILEs (main)
$
```

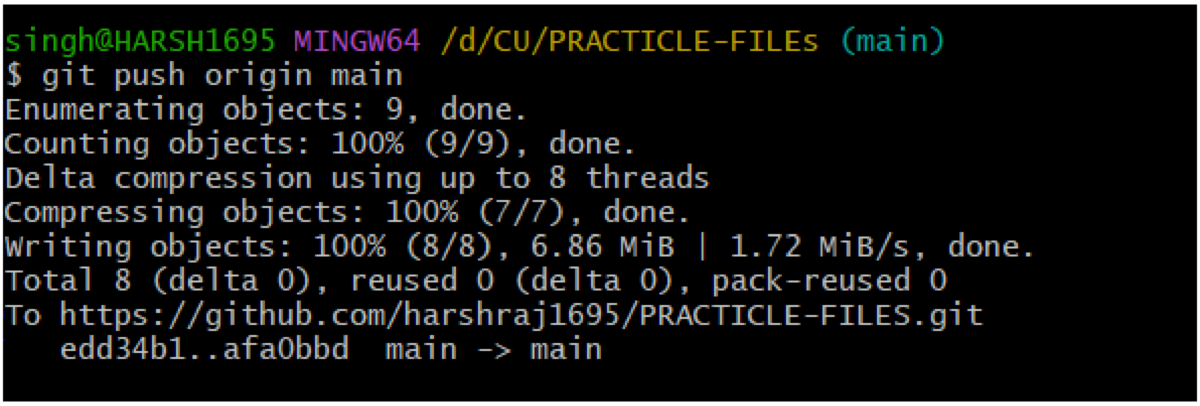
3. Add all the files to the Directory and add to the staging area and commit it using the given two commands.

“git add .”   and “git commit -m”message””



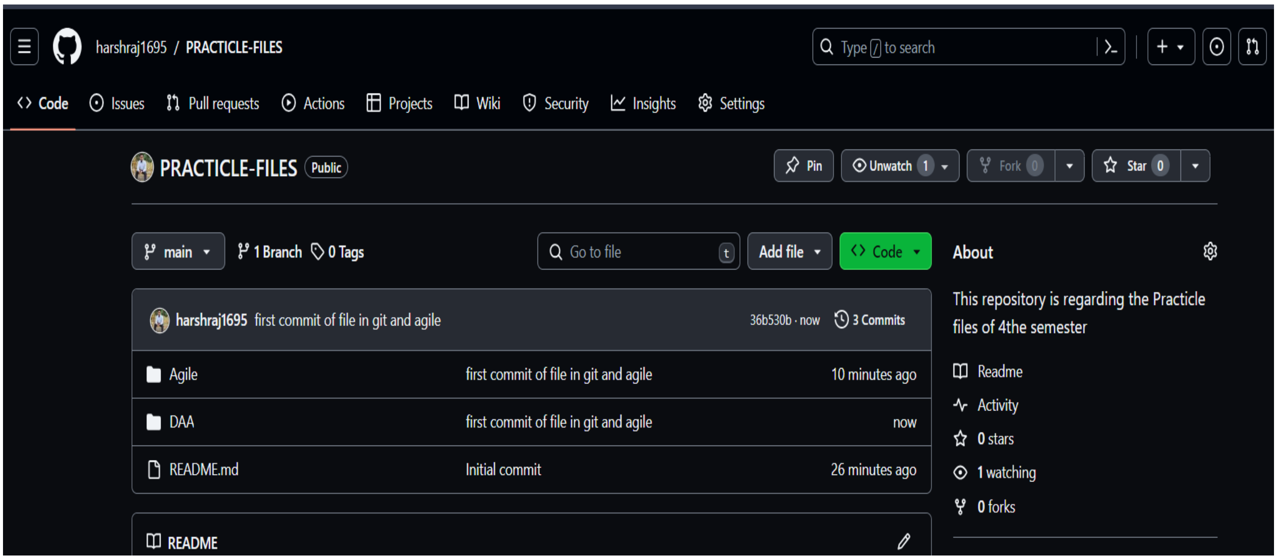
```
singh@HARSH1695 MINGW64 /d/CU/PRACTICLE-FILES (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Agile/
        DAA/

nothing added to commit but untracked files present (use "git add" to track)

singh@HARSH1695 MINGW64 /d/CU/PRACTICLE-FILES (main)
$ git add .

singh@HARSH1695 MINGW64 /d/CU/PRACTICLE-FILES (main)
$ git commit -m "first commit of file in git and agile"
[main afa0bbd] first commit of file in git and agile
 7 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Agile/agile 1.pdf
 create mode 100644 Agile/agile2.pdf
 create mode 100644 Agile/agile3.pdf
 create mode 100644 DAA/practile1.c
 create mode 100644 DAA/practile2.c
 create mode 100644 DAA/practile3.c
 create mode 100644 DAA/practlie4.c
```

4. Push the changes in the remote repository by using “git push origin main”



```
singh@HARSH1695 MINGW64 /d/CU/PRACTICLE-FILES (main)
$ git push origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 6.86 MiB | 1.72 MiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/harshraj1695/PRACTICLE-FILES.git
   edd34b1..afa0bbd  main -> main
```

5. Verify the changes on github.

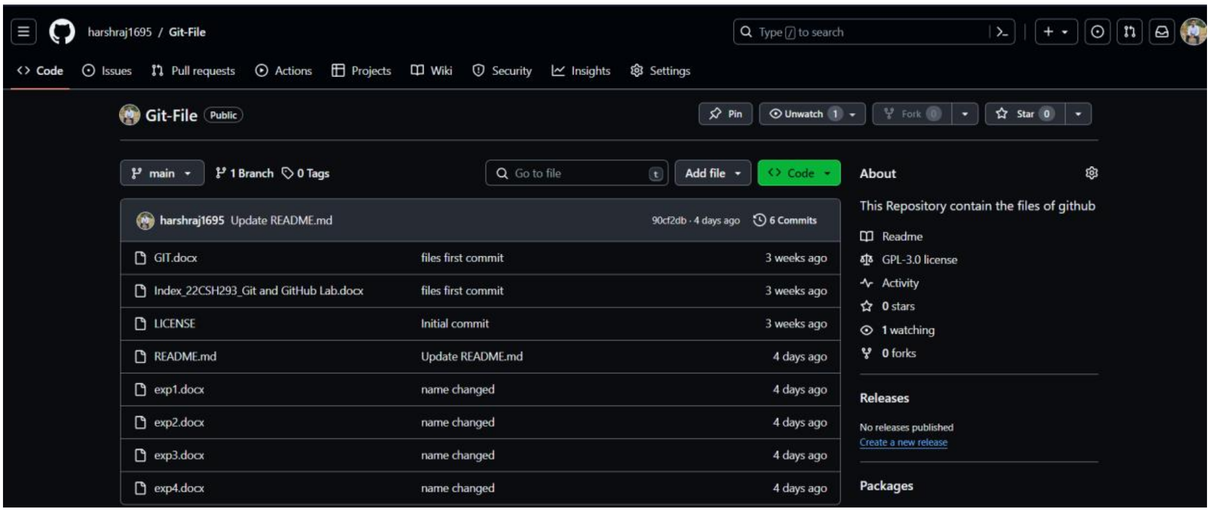


harshraj1695 first commit of file in git and agile 36b530b · now 3 Commits

Agile	first commit of file in git and agile	10 minutes ago
DAA	first commit of file in git and agile	now
README.md	Initial commit	26 minutes ago

This repository is regarding the Practicle files of 4the semester

6. Follow the similar steps to create and push into local repositories of Git file.



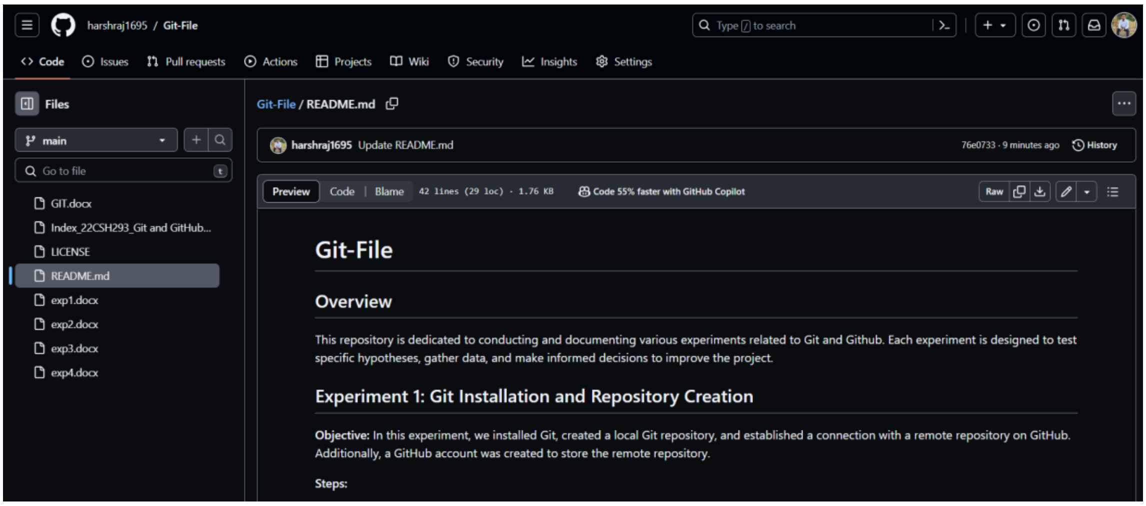
harshraj1695 Update README.md 90cf2db · 4 days ago 6 Commits

GIT.docx	files first commit	3 weeks ago
Index_22CSH293_Git and GitHub Lab.docx	files first commit	3 weeks ago
LICENSE	Initial commit	3 weeks ago
README.md	Update README.md	4 days ago
exp1.docx	name changed	4 days ago
exp2.docx	name changed	4 days ago
exp3.docx	name changed	4 days ago
exp4.docx	name changed	4 days ago

This Repository contain the files of github

7. Edit the readme.md file for the repository to give description about the projects.

8. Edit the "Git File " readme first



harshraj1695 Update README.md 76e0733 · 9 minutes ago

42 lines (29 loc) · 1.76 KB

Git-File

Overview

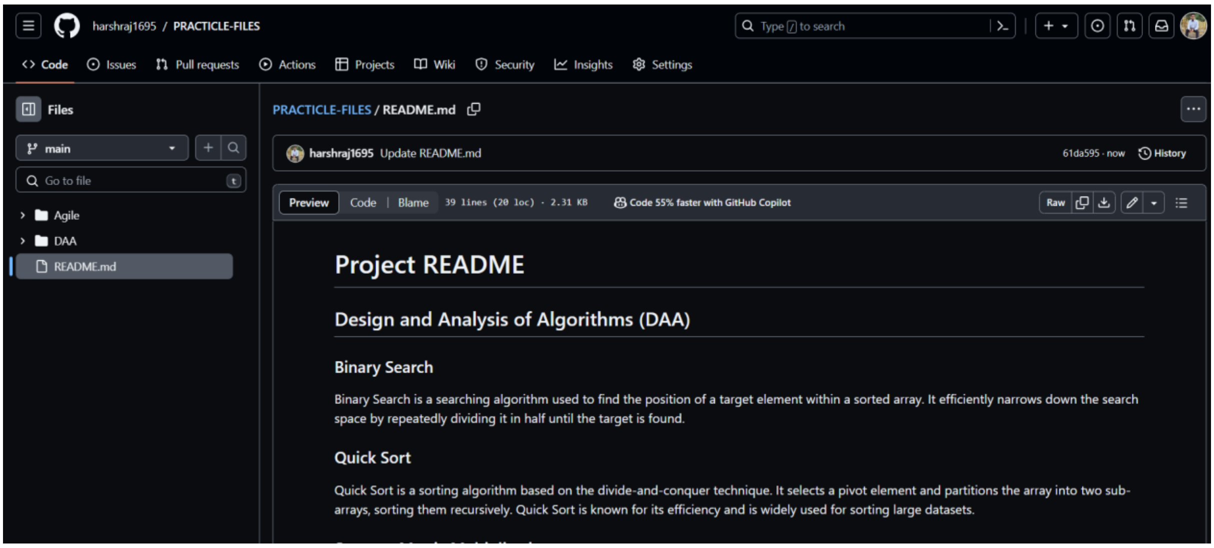
This repository is dedicated to conducting and documenting various experiments related to Git and Github. Each experiment is designed to test specific hypotheses, gather data, and make informed decisions to improve the project.

Experiment 1: Git Installation and Repository Creation

Objective: In this experiment, we installed Git, created a local Git repository, and established a connection with a remote repository on GitHub. Additionally, a GitHub account was created to store the remote repository.

Steps:

9. Edit the Practicle file readme in the same manner.



Project README

Design and Analysis of Algorithms (DAA)

Binary Search

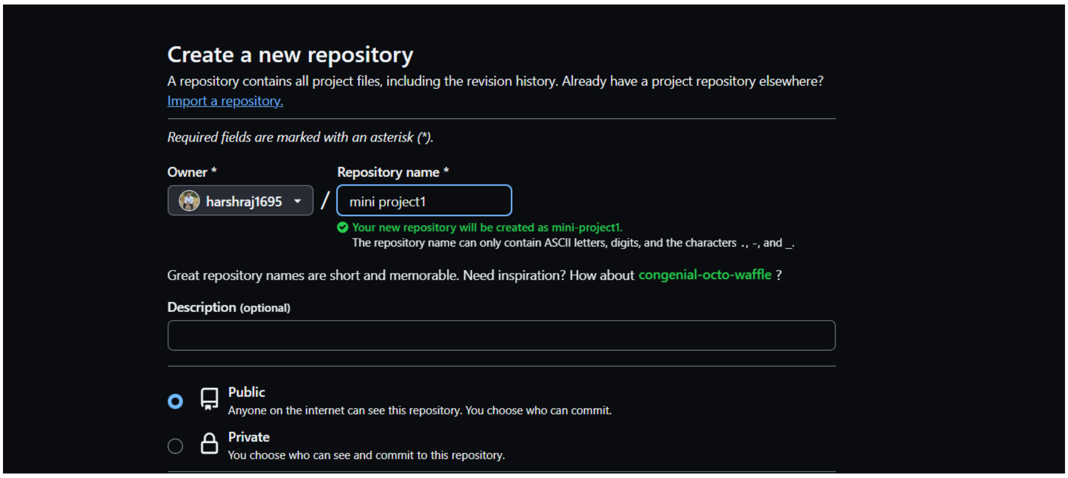
Binary Search is a searching algorithm used to find the position of a target element within a sorted array. It efficiently narrows down the search space by repeatedly dividing it in half until the target is found.

Quick Sort

Quick Sort is a sorting algorithm based on the divide-and-conquer technique. It selects a pivot element and partitions the array into two sub-arrays, sorting them recursively. Quick Sort is known for its efficiency and is widely used for sorting large datasets.

Task 1: Create a Project for the college mini Project and add a repository to it and add all assign all the group members to it.

1. Create a repository in the github name "Project" **.**



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * harshraj1695 / Repository name * mini project1

Your new repository will be created as mini-project1.
The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

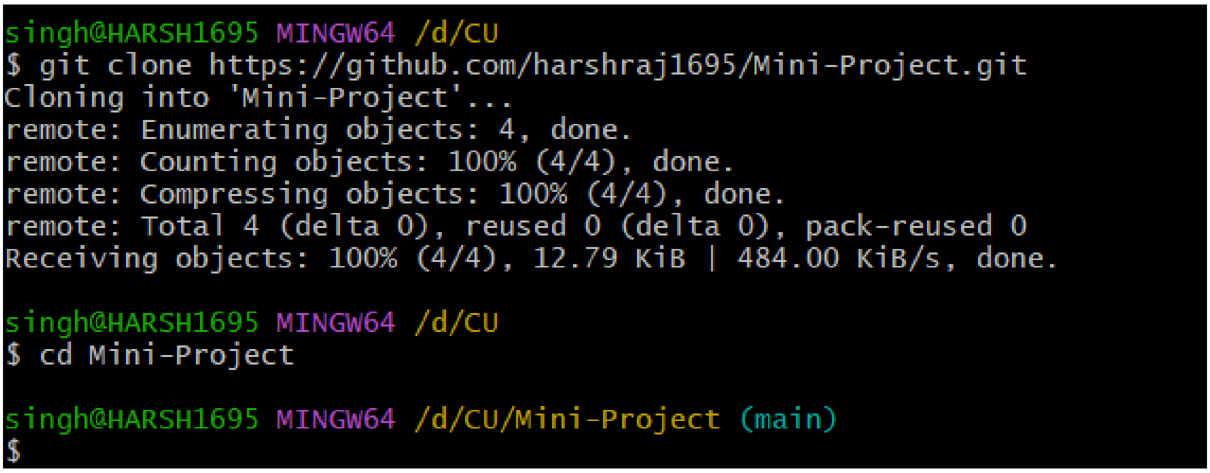
Great repository names are short and memorable. Need inspiration? How about congenial-octo-waffle ?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

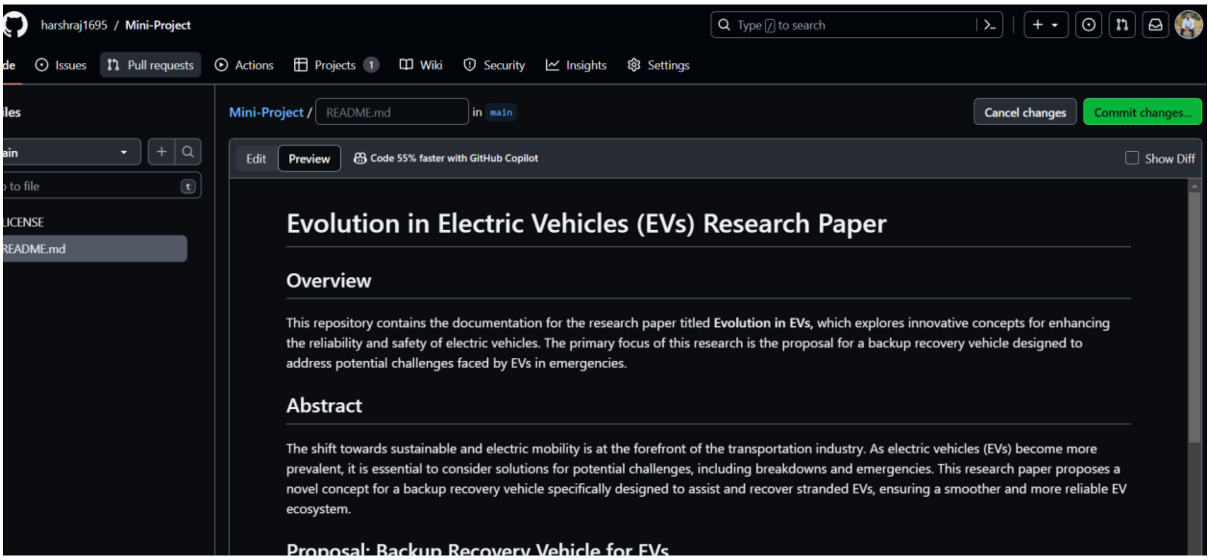
○ Private
You choose who can see and commit to this repository.

2. Clone this repository in the local machine using git clone command and change to that directory using the "cd " command.



```
singh@HARSH1695 MINGW64 /d/CU
$ git clone https://github.com/harshraj1695/Mini-Project.git
Cloning into 'Mini-Project'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), 12.79 KiB | 484.00 KiB/s, done.

singh@HARSH1695 MINGW64 /d/CU
$ cd Mini-Project

singh@HARSH1695 MINGW64 /d/CU/Mini-Project (main)
$
```

3. Change the Readme.md file to give description regarding the project.



Evolution in Electric Vehicles (EVs) Research Paper

Overview

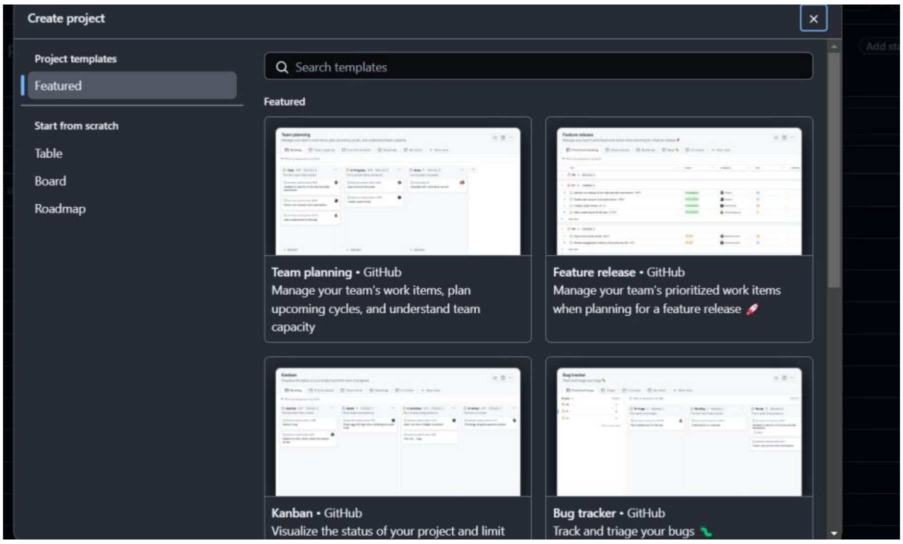
This repository contains the documentation for the research paper titled **Evolution in EVs**, which explores innovative concepts for enhancing the reliability and safety of electric vehicles. The primary focus of this research is the proposal for a backup recovery vehicle designed to address potential challenges faced by EVs in emergencies.

Abstract

The shift towards sustainable and electric mobility is at the forefront of the transportation industry. As electric vehicles (EVs) become more prevalent, it is essential to consider solutions for potential challenges, including breakdowns and emergencies. This research paper proposes a novel concept for a backup recovery vehicle specifically designed to assist and recover stranded EVs, ensuring a smoother and more reliable EV ecosystem.

Proposal: Backup Recovery Vehicle for EVs

4. In the same repository Click on the "project" and select "new project" and select the template of the project.



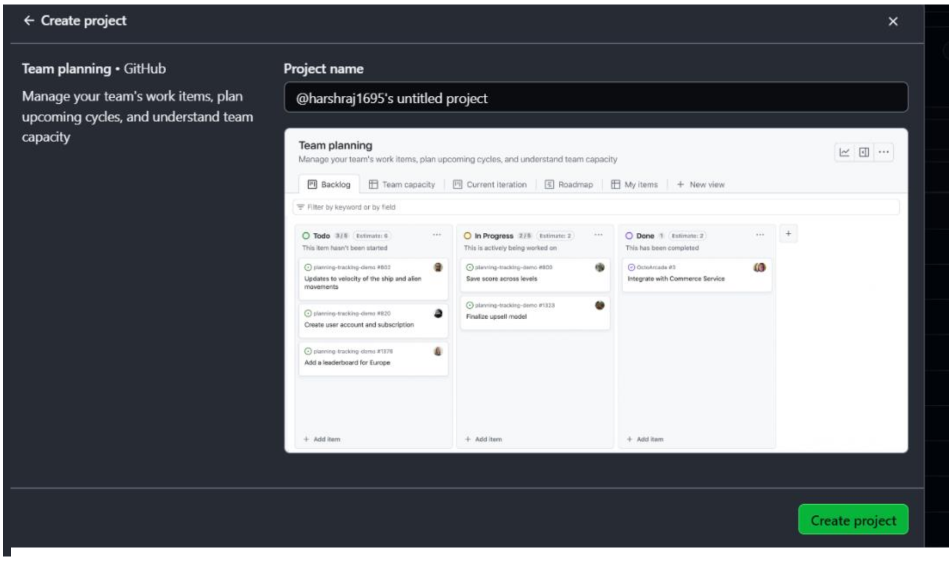
Team planning • GitHub
Manage your team's work items, plan upcoming cycles, and understand team capacity

Feature release • GitHub
Manage your team's prioritized work items when planning for a feature release

Kanban • GitHub
Visualize the status of your project and limit

Bug tracker • GitHub
Track and triage your bugs

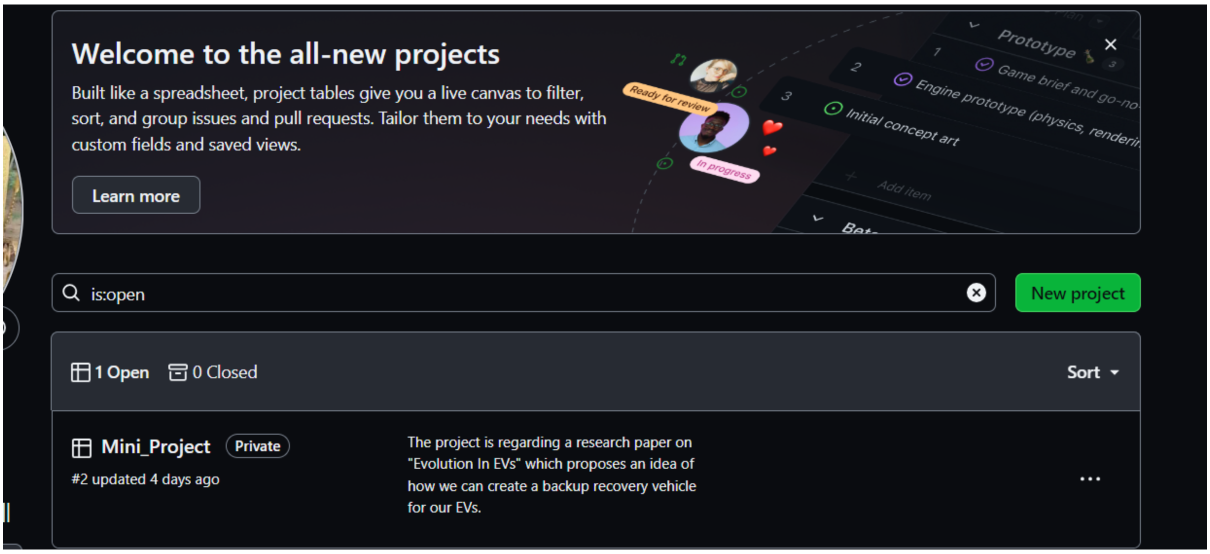
5. Select "Team planing" and click "Create project ".



Team planning • GitHub
Manage your team's work items, plan upcoming cycles, and understand team capacity

Project name: @harshraj1695's untitled project

6. Now go to the Setting and describe the project and click commit changes.



Welcome to the all-new projects

Built like a spreadsheet, project tables give you a live canvas to filter, sort, and group issues and pull requests. Tailor them to your needs with custom fields and saved views.

is:open

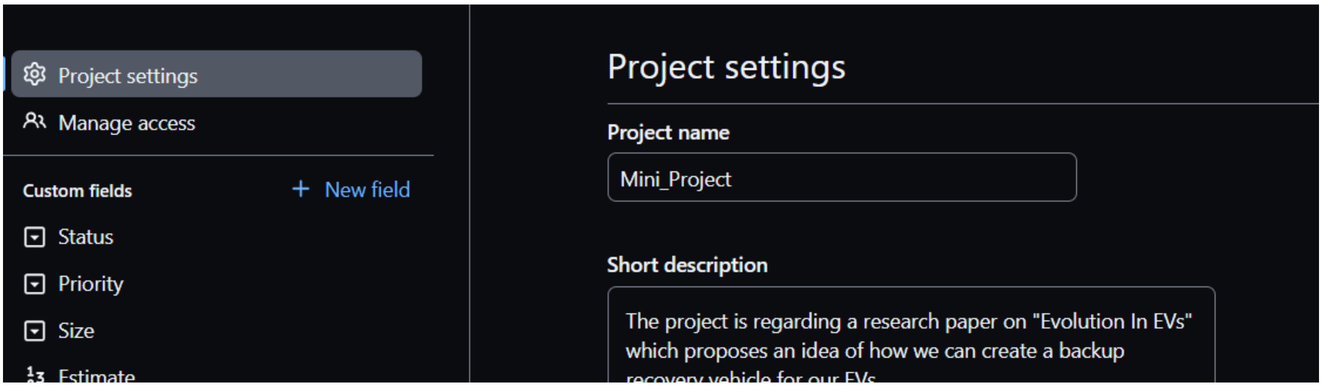
1 Open 0 Closed Sort

Mini_Project Private

#2 updated 4 days ago

The project is regarding a research paper on "Evolution In EVs" which proposes an idea of how we can create a backup recovery vehicle for our EVs.

7. Click on “mini-project ” -> go to settings -> click on manage access.



Project settings

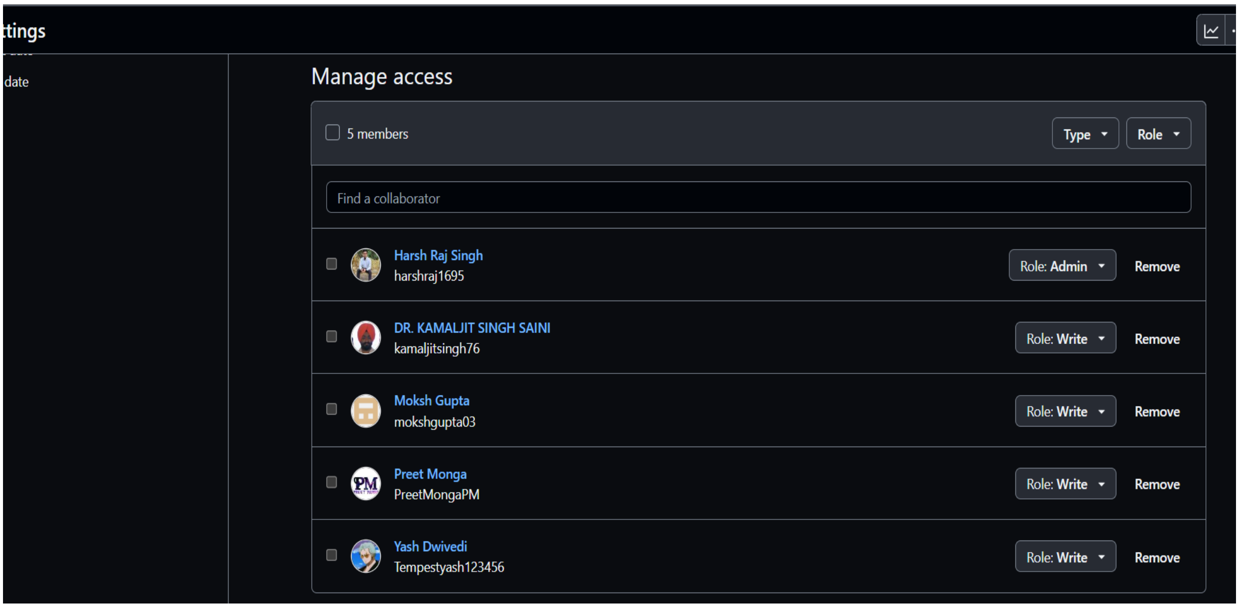
Project name

Mini_Project

Short description

The project is regarding a research paper on "Evolution In EVs" which proposes an idea of how we can create a backup recovery vehicle for our EVs

8. Now assign access to the team members.



Manage access

□ 5 members Type Role

Find a collaborator

□ Harsh Raj Singh harshraj1695 Role: Admin Remove

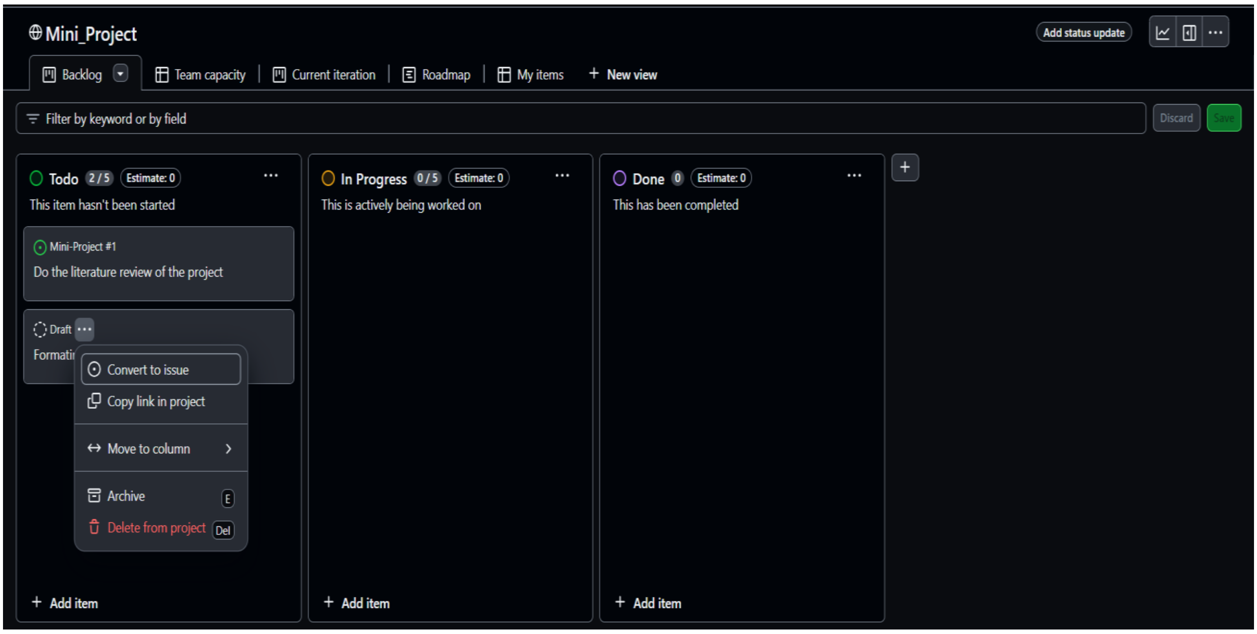
□ DR. KAMALJIT SINGH SAINI kamaljitsingh76 Role: Write Remove

□ Moksh Gupta mokshgupta03 Role: Write Remove

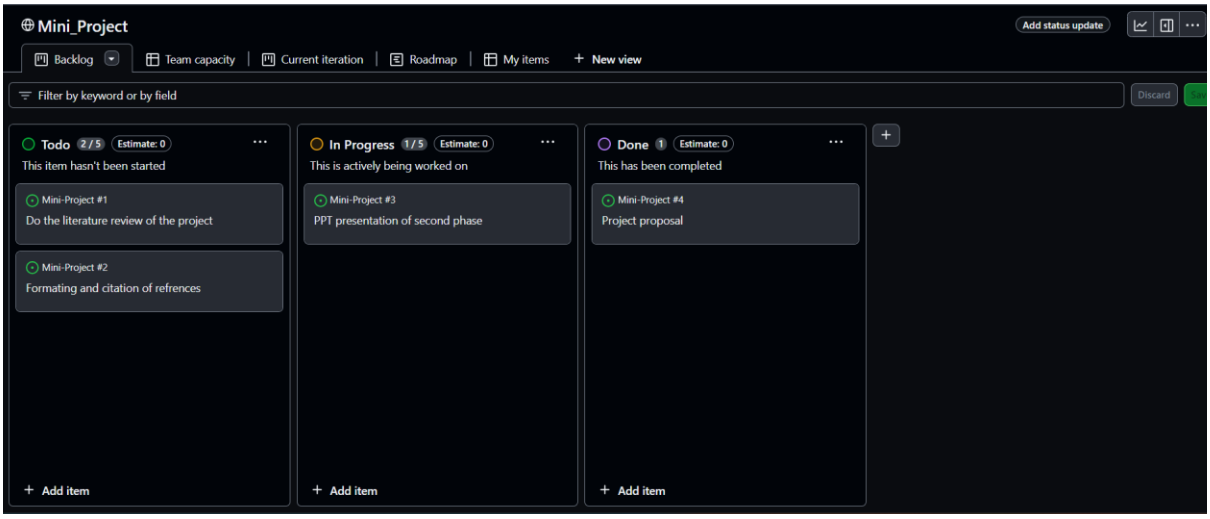
□ Preet Monga PreetMongaPM Role: Write Remove

□ Yash Dwivedi Tempestyash123456 Role: Write Remove

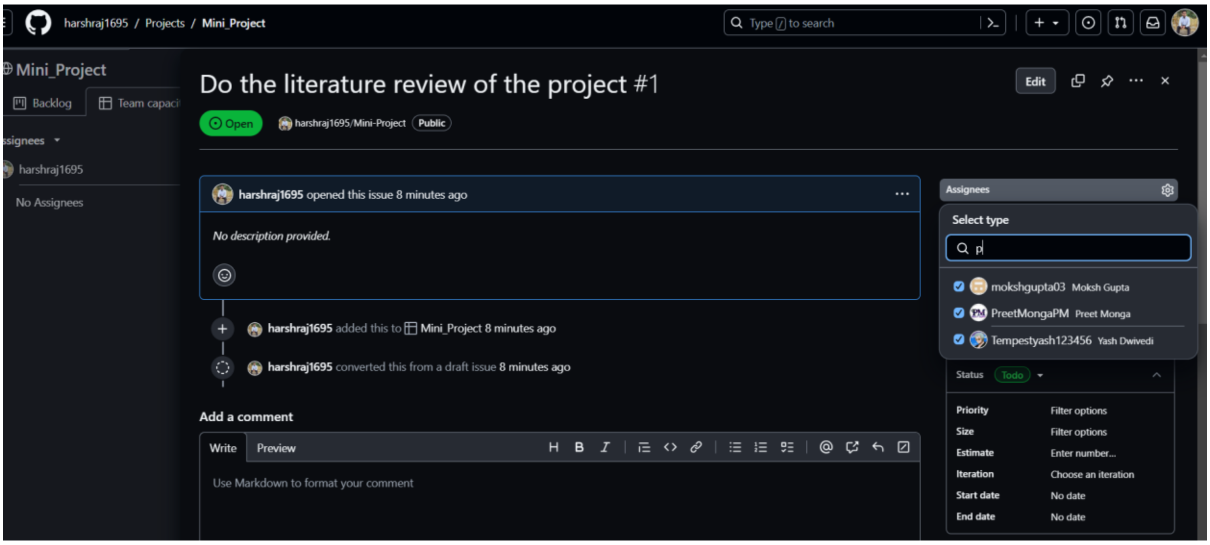
9. Now Go to the project dashboard and create issues in the project.



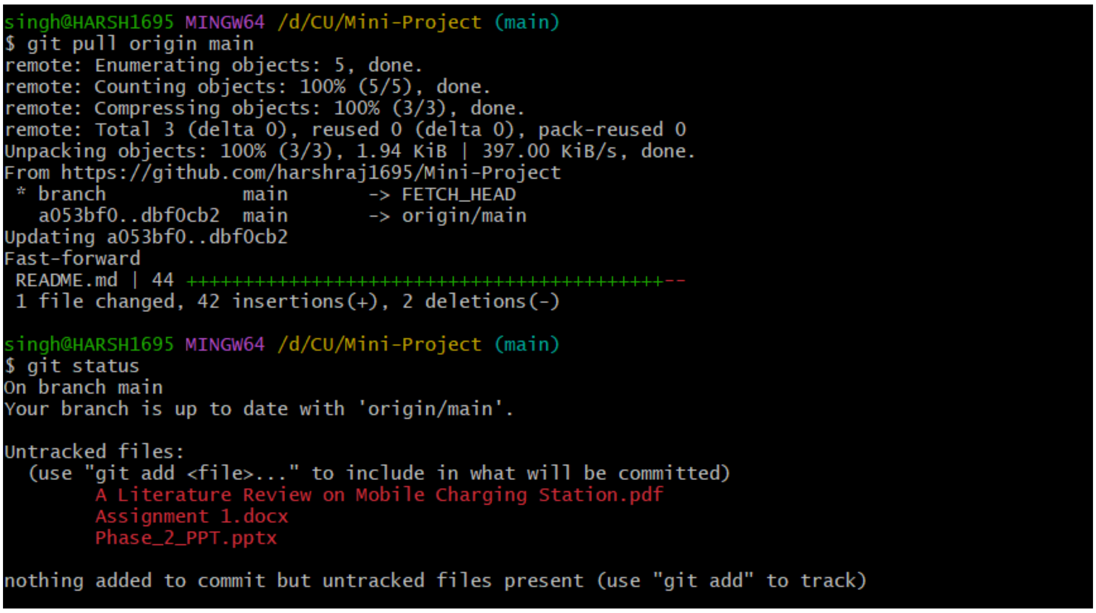
10. List all the issues in the Todo, In Progress and Done section



11. Assign the issues to the group members.

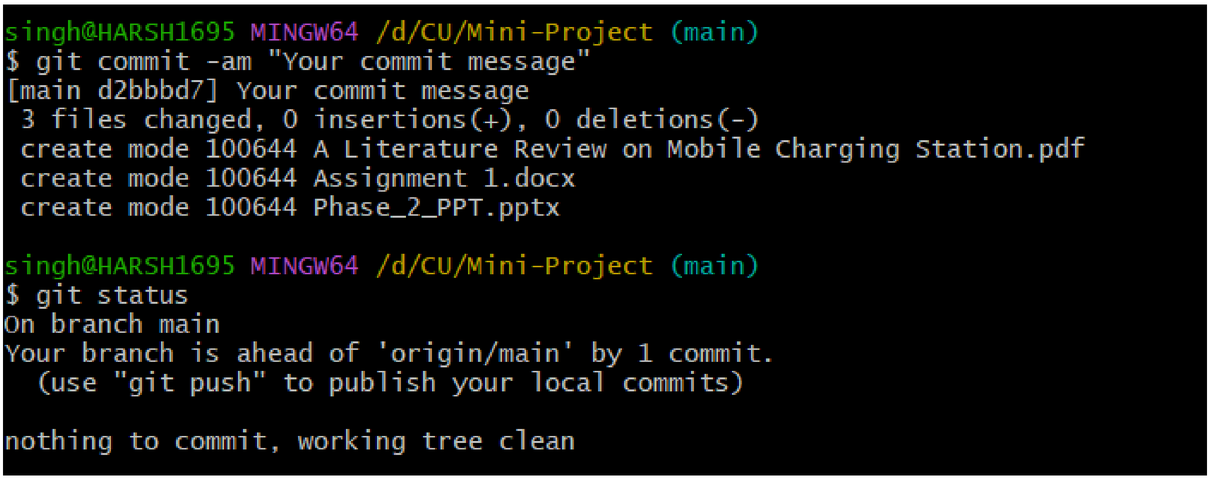

harshraj1695 / Projects / Mini_Project
Type / to search
Mini_Project
Backlog
Team capaci
ssignees
harshraj1695
No Assignees
Do the literature review of the project #1
Edit
Open
harshraj1695/Mini-Project Public
harshraj1695 opened this issue 8 minutes ago
No description provided.
harshraj1695 added this to Mini_Project 8 minutes ago
harshraj1695 converted this from a draft issue 8 minutes ago
Add a comment
Write
Preview
Use Markdown to format your comment
Assignees
Select type
p
mokshgupta03 Moksh Gupta
PreetMongaPM Preet Monga
Tempestyash123456 Yash Dwivedi
Status Todo
Priority Filter options
Size Filter options
Estimate Enter number...
Iteration Choose an iteration
Start date No date
End date No date

12. Now go to git bash and pull all changes in the local repository.



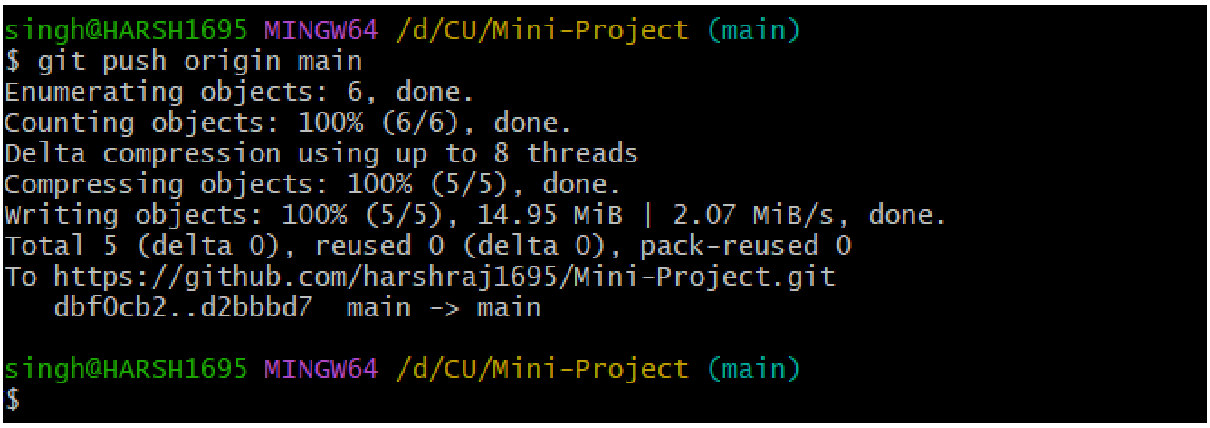
```
singh@HARSH1695 MINGW64 /d/CU/Mini-Project (main)
$ git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 1.94 KiB | 397.00 KiB/s, done.
From https://github.com/harshraj1695/Mini-Project
 * branch            main       -> FETCH_HEAD
   a053bf0..dbf0cb2  main       -> origin/main
Updating a053bf0..dbf0cb2
Fast-forward
 README.md | 44 ++++++++++++++++++++++++++++++++++++++++++--
 1 file changed, 42 insertions(+), 2 deletions(-)

singh@HARSH1695 MINGW64 /d/CU/Mini-Project (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        A Literature Review on Mobile Charging Station.pdf
        Assignment 1.docx
        Phase_2_PPT.pptx

nothing added to commit but untracked files present (use "git add" to track)
```

13. Now perform add and commit on the folders.



```
singh@HARSH1695 MINGW64 /d/CU/Mini-Project (main)
$ git commit -am "Your commit message"
[main d2bbbd7] Your commit message
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 A Literature Review on Mobile Charging Station.pdf
 create mode 100644 Assignment 1.docx
 create mode 100644 Phase_2_PPT.pptx

singh@HARSH1695 MINGW64 /d/CU/Mini-Project (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

14. Now push the changes into the remote repository.



```
singh@HARSH1695 MINGW64 /d/CU/Mini-Project (main)
$ git push origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 14.95 MiB | 2.07 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/harshraj1695/Mini-Project.git
   dbf0cb2..d2bbbd7  main -> main

singh@HARSH1695 MINGW64 /d/CU/Mini-Project (main)
$
```